Optimization Models for Congestion Mitigation in Virtual Networks

Jocelyne Elias*, Fabio Martignon†, Stefano Paris* and Jianping Wang‡

*LIPADE
Université Paris Descartes
{jocelyne.elias, stefano.paris}@parisdescartes.fr

†LRI
University of Paris-Sud
Institut Universitaire de France (IUF)
fabio.martignon@lri.fr

‡Department of Computer Science
City University of Hong Kong
jianwang@cityu.edu.hk

***Abstract*—Virtualization of network functions and services can significantly reduce capital and operational expenditures of telecommunication operators through the sharing of a single network infrastructure. However, the utilization of the same resources can increase their congestion due to the spatio-temporal correlation of traffic demands and computational loads.**

In this paper, we propose novel *orchestration mechanisms* to optimally control and reduce the resource congestion of a physical infrastructure based on the NFV paradigm. In particular, we formulate the network functions composition problem as a non-linear optimization model to accurately capture the congestion of the physical resources. In order to meet both efficiency and load balancing goals of the physical operator, we introduce two variants of such model to minimize the *total* and the *maximum* congestion in the network. Our models allow us to efficiently compute the optimal solution in a short computing time. Numerical results, obtained with real ISP topologies and network instances, show that the proposed approach represents an efficient and practical solution to control the congestion in virtual networks. Furthermore, they indicate that a holistic approach that optimizes the virtual system by jointly considering all elements/components would further improve the performance.

***Index Terms*—Network Functions Virtualization, Congestion Control, Pricing, Non-linear Optimization.**

I. Introduction

Nowadays, telecommunication networks consist of a huge set of property hardware appliances and software systems that implement the necessary communication and computation infrastructure to provide high-quality services to their final customers. Nonetheless, seamlessly integrating a new service or network function represents a challenging task due to the high complexity level attained by the infrastructure as well as the low degree of flexibility of the network equipment, which is usually designed and optimized for specific tasks. As a consequence, network operators have implemented *virtualized solutions* that enable the sharing of general-purpose resources to increase the flexibility of their infrastructures that, in turn, can be used to support heterogeneous services. One of the first successful virtualization attempts is represented by cloud computing, which increases the resource availability and utilization through a middleware that hides the underlying infrastructure's complexity.

To improve the interoperability of different virtualization technologies and further reduce operational costs, network operators and equipment manufacturers are consolidating the virtualization technology, designing new standards for the implementation of network functions onto reprogrammable and reconfigurable network devices like high volume servers, switches, storage systems and base stations. Such a specification, which is called *Network Functions Virtualization* (NFV) [1], enables the seamless integration and execution of a new network function or service onto a large range of network equipment. NFV brings about several benefits for network operators, such as reduced CAPEX and OPEX (CAPital and OPerational EXpenditure), low time-to-market for the development of new network services, higher flexibility to scale up and down the services according to users' demand, simple and cheap testing of new services, and lower risks for the launch of innovative services, since there is no need to purchase new and expensive devices. Furthermore, NFV simplifies the spatial and temporal deployment of important operational and management tasks, such as traffic analysis, billing, sampling and verification, defining the best set of devices for their execution.

Nonetheless, these technologies pose new challenges to the optimal management of the entire infrastructure, since they deeply modify the classical architecture and utilization of networking systems. Indeed, the sharing of the physical infrastructure among multiple virtual operators as well as the simple configuration of network functions and services rise several problems that can lead to an unfair use of the available system resources, unless management procedures are deployed to dynamically control the resource utilization.

In this paper, we propose novel orchestration mechanisms to optimally control and mitigate the resource congestion of a physical infrastructure based on the NFV paradigm. We formulate the problem as a non-linear optimization model that accurately captures the congestion of physical network resources, and permits to dynamically control traffic flows and system configurations in order to prevent the congestion of network resources. More specifically, we consider two models to minimize the total and the maximum congestion of the physical infrastructure in order to cover the widest spectrum of physical operator's goals. Indeed, the minimization of the total congestion implements a load balancing approach that exploits the maximum number of available resources to fairly assign network slices to all virtual operators. In contrast, the minimization of the maximum congestion results in a configuration that efficiently uses network resources, aggregating the maximum number of requests of virtual operators.

Numerical results show that the proposed models significantly decrease network congestion, thus representing a very promising approach for operators to manage network resources in an efficient and dynamic fashion.

The paper is structured as follows. Section II discusses related work. Section III introduces the network model as well as the assumptions considered in our work. Section IV formulates the congestion mitigation problem of virtual networks as a non-linear optimization problem, while Section V illustrates and analyzes numerical results that show the efficiency and validity of our approaches. Finally, concluding remarks are discussed in Section VI.

II. Related Work

Emerging paradigms like Software-Defined Networks (SDN) and Network Functions Virtualization (NFV) [1] are envisioned to help making the innovation cycles of network and service features faster and simpler. These two paradigms will therefore contribute to reduce the ossification of Internet and Telecom networks, which is creating several difficulties for Service Providers and Network Operators to develop and deploy innovative network functionalities, services and management policies, which are essential to benefit from the increasing dynamicity of the ICT markets. Recent surveys and discussions on network virtualization can be found in [2], [3], [4].

In particular, the virtualization paradigm stems out as a cost-effective strategy to efficiently exploit a shared physical hardware infrastructure. In the context of network virtualization, the embedding problem has been thorougly investigated in the community, and multiple algorithms have already been presented [5], [6], [7], [8], [9].

The Virtual Network Embedding problem consists in finding a mapping between a set of requests for virtual network resources and the available underlying physical infrastructure (the *substrate*), ensuring that some given performance requirements (on nodes and links) are guaranteed. The problem is known to be NP-hard, since it can be reduced to the multi-way separator problem, and for this reason heuristic algorithms have been proposed [10].

Botero et al. in [9] tackle the problem of virtual resources consolidation form the energy efficiency point of view. The authors formulate a mixed integer linear programming (MILP) model to understand the potential benefits that can be achieved by packing many different virtual tasks on the same physical infrastructure, showing up to 30% in energy savings. The work [11] presents a solution for the resilient deployment of network functions, using OpenStack for the design and implementation of the proposed service orchestrator mechanism.

In [12], authors present an allocation mechanism based on auction theory to select the most remunerative virtual network requests according to QoS requirements and physical constraints. A more general virtual topology embedding problem is presented in [13], where the underlying physical infrastructure is managed by two types of providers: cloud providers and transit network providers.

To the best of our knowledge, only few works focus on congestion control in virtual networks. On the contrary, in this paper we study the effects on the network congestion of services composition in NFV-based infrastructures, in order to derive numerical bounds on the congestion reduction that can be achieved by deploying virtualization mechanisms.

III. Network Model

This section presents the network model and assumptions we adopt in the design of our mechanism for controlling and mitigating resource congestion in virtual networks.

We consider a physical network infrastructure managed by a single operator composed of a set $\mathcal{N}$ of general purpose nodes and a set $\mathcal{L}$ of directed links. Therefore, the topology of the network infrastructure is represented as a weighted directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$. The operator adopts the Network Functions Virtualization (NFV) approach for providing access to its physical resources, since through the virtualization of network functions, virtual operators can share physical resources implementing their own network services independently of each other and the underlying technology. Therefore, the set $\mathcal{U}$ of virtual operators can design their own virtual network using the basic network functions and network equipment provided by the physical operator to compose their services.

Each virtual operator[1] $u \in \mathcal{U}$ defines its demand b_u by specifying the source and destination nodes as well as the amount of data traffic r^u that is transmitted between them, that is $b^u = \{s^u, d^u, r^u\}$. In our vision, the virtual operator also provides a list of processing nodes ($\mathcal{P}_u$) through which a fraction of the data traffic $w_j^u \in [0, 1]$ must be routed. These nodes are used to perform intensive computational tasks, like traffic analysis and deep packet inspection (e.g., firewalls, intrusion detection systems and other in-the-cloud middlebox services [14]) and to facilitate key operations like billing, sampling and verification. For example, a VO can verify the service provided by the physical operators based on the performance of packets passing through a specified node of the set $\mathcal{P}_u$).

In the Mixed Integer Linear Programming (MILP) formulation of our congestion minimization problem for NFV-based infrastructures, we define flow variables $x_{ij}^{us} \geq 0$, which denote the amount of traffic of VO u passing through link $(i; j)$ and transmitted using service s. Furthermore, we introduce binary decision variables $y_{ij}^{us} \in \{0, 1\}$ to identify the services that are used for the transmission of VO traffic and avoiding traffic splitting over multiple services ($y_{ij}^{us} = 1$ indicates that service s on link $(i; j)$ is selected to transmit the data traffic of user u).

The *congestion cost function*, which we use for achieving the best configuration of network functions, depends on the network congestion in order to fully exploit the physical infrastructure and guarantee, at the same time, a high Quality of Service (QoS). This function is commonly used in the

[1] User and Virtual Operator (VO) are used interchangeably throughout the paper.

literature [15], [16]. More specifically, for each link $(i;j) \in \mathcal{L}$, we consider an *increasing* and *convex* function as follows:

$$J_{ij}\left(\sum_{s\in\mathcal{S}_{ij}} \frac{\sum_{u\in\mathcal{U}} x_{ij}^{us}}{B_{ij}^{s}}\right) = a_{ij} + b_{ij}\left(\sum_{s\in\mathcal{S}_{ij}} \frac{\sum_{u\in\mathcal{U}} x_{ij}^{us}}{B_{ij}^{s}}\right)^{\tau} \quad (1)$$

where x_{ij}^{us} is the traffic flow of VO u passing through link $(i;j)$ and transmitted using the service s, while B_{ij}^{s} represents the bandwidth assigned to service s on link $(i;j)$. The coefficients a_{ij} and b_{ij} are two positive numbers, which are used to model the overhead caused by the virtualization technology, and τ is a positive integer greater than 1.

We observe that Equation (1) represents only the congestion cost experienced by the operator due to the congestion on a single link. Therefore, the total cost J incurred by the physical operator due to the overall network congestion can be expressed as follows, summing the cost over all links:

$$J = \sum_{(i;j)\in\mathcal{L}}\left[J_{ij}\left(\sum_{s\in\mathcal{S}_{ij}} \frac{\sum_{u\in\mathcal{U}} x_{ij}^{us}}{B_{ij}^{s}}\right)\right]. \quad (2)$$

IV. Optimal Congestion Mitigation for Virtual Networks

In this section, we consider two variants of a non-linear integer programming model of the Congestion Mitigation for Virtual Networks (CMVN) optimization problem. The first formulation aims at minimizing the *total network congestion*, whereas the second alternative relieves the congestion on the bottleneck link (i.e., it minimizes the *most congested link*).

A. Total Congestion Formulation

The total network congestion is defined as the sum over all links and transmission services of a convex function that depends on the link utilization, as formulated in Equation (2). According to such definition, the problem of optimally minimizing the total network congestion (which we simply call MinTot-CMVN) can be therefore formulated as follows:

$$\min \quad J = \sum_{(i;j)\in\mathcal{L}} J_{ij} \quad (3)$$

s.t.

$$\sum_{\substack{i\in\mathcal{N}:\\(j;i)\in\mathcal{L}}}\sum_{s\in\mathcal{S}_{ji}} x_{ji}^{us} - \sum_{\substack{i\in\mathcal{N}:\\(i;j)\in\mathcal{L}}}\sum_{s\in\mathcal{S}_{ij}} x_{ij}^{us} = r^{u} \quad \forall u\in\mathcal{U}, \forall j\in\mathcal{N}: j = s^{u} \quad (4)$$

$$\sum_{\substack{i\in\mathcal{N}:\\(j;i)\in\mathcal{L}}}\sum_{s\in\mathcal{S}_{ji}} x_{ji}^{us} - \sum_{\substack{i\in\mathcal{N}:\\(i;j)\in\mathcal{L}}}\sum_{s\in\mathcal{S}_{ij}} x_{ij}^{us} = 0 \quad \forall u\in\mathcal{U}, \forall j\in\mathcal{N}: j \neq s^{u} \land j \neq d^{u} \quad (5)$$

$$\sum_{\substack{i\in\mathcal{N}:\\(j;i)\in\mathcal{L}}}\sum_{s\in\mathcal{S}_{ji}} x_{ji}^{us} - \sum_{\substack{i\in\mathcal{N}:\\(i;j)\in\mathcal{L}}}\sum_{s\in\mathcal{S}_{ij}} x_{ij}^{us} = -r^{u} \quad \forall u\in\mathcal{U}, \forall j\in\mathcal{N}: j = d^{u} \quad (6)$$

$$\sum_{u\in\mathcal{U}} x_{ij}^{us} \leq B_{ij}^{s} \quad \forall (i;j)\in\mathcal{L}, \forall s\in\mathcal{S}_{ij} \quad (7)$$

$$\sum_{i\in\mathcal{N}}\sum_{s\in\mathcal{S}_{ij}} x_{ij}^{us} = w_{j}^{u} r^{u} \quad \forall u\in\mathcal{U}, j\in\mathcal{P}_{u} \quad (8)$$

$$\sum_{s\in\mathcal{S}_{ij}} y_{ij}^{us} = 1 \quad \forall (i,j)\in\mathcal{L}, u\in\mathcal{U} \quad (9)$$

$$x_{ij}^{us} \leq y_{ij}^{us} r^{u} \quad \forall (i,j)\in\mathcal{L}, u\in\mathcal{U}, s\in\mathcal{S}_{ij} \quad (10)$$

$$x_{ij}^{us} \geq 0 \quad \forall (i,j)\in\mathcal{L}, u\in\mathcal{U}, s\in\mathcal{S}_{ij} \quad (11)$$

$$y_{ij}^{us} \in \{0,1\} \quad \forall (i,j)\in\mathcal{L}, u\in\mathcal{U}, s\in\mathcal{S}_{ij}. \quad (12)$$

Objective function (3) minimizes the overall network congestion, defined according to the model described in Section III. Constraints (4)-(6) define the flow balance at node $j \in \mathcal{N}$ for the data traffic demand of user u. Specifically, terms $\sum x_{ij}^{us}$ and $\sum x_{ji}^{us}$ represent the total incoming and outgoing traffic flows, respectively.

The set of constraints (7) ensures that the total traffic routed on a link established between two devices i and j using the transmission service s does not exceed the bandwidth assigned by the operator to the service s, which is denoted by B_{ij}^{s}. The set of constraints (8) forces the fraction of data traffic that must be processed or analyzed, w_{j}^{u}, to pass through the processing nodes ($\mathcal{P}_{u}$) selected by the virtual operator. Therefore, the physical operator can select the set of physical nodes that minimize the congestion to perform the computational tasks requested and developed by the virtual operator u for its data traffic.

Constraints (9) force the utilization of a single service for the user traffic transmission on link $(i;j)$, while (10) ensures that the traffic flow routed through that link does not exceed the user demand, r^{u}. The first set of constraints is added to our model in order to reduce the system complexity. Indeed, splitting the user data traffic on a link among multiple transmission services requires sophisticated scheduling procedures that increase the system complexity. Finally, constraints (11) ensure the positiveness of the flow variables, while (12) ensure the integrality of the binary decision variables.

Similarly to classical traffic engineering techniques proposed for wired networks [17], our work is based on the idea of using a non-linear increasing and convex function to strongly penalize network configurations that intensively use only few links. However, our work is unique with respect to the underlying traffic and network models, which accurately capture the flexibility and reconfigurability features of infrastructures based on the NFV technology. In particular, differently from virtual embedding problems like [5], [18], our model considers different transmission services for each link (e.g., MAC protocols and scheduling policies) and the nonlinear effect on the link's congestion and capacity degradation caused by scheduling mechanisms when the contention level increases. Such an effect is typical in communication systems based on resource sharing, which show an exponential response time. Finally, besides the accurate modeling of NFV transmission services, our proposed orchestration mechanism provides a certain degree of flexibility for the placement of network services (like billing, caching, traffic sampling and verification) that require the execution of complex functions on physical machines.

B. Worst Congestion Formulation

In specific scenarios, the operator of the physical infrastructure might want to reduce the traffic flowing through the *most congested link* for load balancing purposes and fairly using all network resources. Therefore, in the following we introduce an alternative formulation of the CMVN problem that allows the physical network operator to minimize the worst congestion (we call this model MinMax-CMVN).

To this end, let us define a function U_{cong} that represents an upper bound on the link congestion as:

$$a_{ij} + b_{ij}\left(\sum_{s\in\mathcal{S}_{ij}} \frac{\sum_{u\in\mathcal{U}} x_{ij}^{us}}{B_{ij}^{s}}\right)^{\tau} \leq U_{cong} \tag{13}$$

The MinMax-CMVN model can therefore be defined as:

$$\min \quad U_{cong} \tag{14}$$

s.t.

$$a_{ij} + b_{ij}\left(\sum_{s\in\mathcal{S}_{ij}} \frac{\sum_{u\in\mathcal{U}} x_{ij}^{us}}{B_{ij}^{s}}\right)^{\tau} \leq U_{cong}, \ \forall (i,j)\in\mathcal{L}. \tag{15}$$

with all constraints (4)-(12) introduced in the previous model.

V. Numerical Results

This section presents numerical results that illustrate the validity of the proposed *MinTot-CMVN* and *MinMax-CMVN* models to solve the congestion mitigation problem for virtual networks. More specifically, we test the sensitivity of the proposed models to different parameters like the number of virtual operators, and their requests (i.e., traffic demands, number of services). In particular, we evaluate the impact of these parameters on the performance of the physical operator's network.

We first describe the experimental methodology of our simulations. Then, we analyze and discuss the performance achieved by the proposed algorithms.

A. Experimental Methodology

In order to evaluate the network performance achieved by the two congestion mitigation problems that we design to compose virtual network functions, we formulate the corresponding optimization models in AMPL. Since we consider $\tau = 1$, we use CPLEX as solver. Note, however, that nonlinear optimization solvers like SNOPT can be used to compute the optimal solutions when $\tau > 1$. Furthermore, we implemented a standalone program in C++ that, starting from the definition of real network topologies[2] (either in GML or GraphML), generates the NFV scenario as follows. We extend the *Geant* and *Cogentco* topologies, which contain 40 and 197 nodes connected through 122 and 486 directed links, respectively. The capacity of all links has been normalized to simplify the analysis of the network congestion. Furthermore, we vary the number of transmission services for each link in the range $[2, 4]$ to quantify the control overhead that may be introduced using multiple virtual services. Note that the link capacity has been evenly divided among all transmission services, since we assume the implementation of a round robin scheme for scheduling multiple virtual services.

For each Virtual Operator $u \in \mathcal{U}$, we randomly select the source and destination of the data connection (i.e., s^u and d^u), which represent the ingress and egress points of the VO. The bandwidth demand of every virtual operator is drawn according to both a uniform and a skewed distribution in $[0, 1]$. Regarding the skewed distribution, we divided the set of VOs in three classes assigning connections with different bandwidth requirements, namely *mice*, *normal* and *elephants*. *Normal* and *elephant* data connections require twice (2x) and three times (3x) more bandwidth than *mice*, respectively. Sets sizes and bandwidth requests of the three classes have been computed to generate the same amount of traffic obtained with the uniform distribution. Note that the granularity of our representation for VOs' requests is highly flexible. Indeed, we can represent VOs demand with multiple connections by simply defining a different VO for each pair of ingress/egress points.

Regarding the processing nodes, which may implement important network functionalities such as storage/caching, security, billing, traffic analysis and filtering, we vary their number in the range $[2, 4]$, selecting randomly the network nodes that implement the processing functions for each VO. We assume that all network functions are replicated on all processing nodes for reliability and load balancing purposes. Therefore, we split equally the portion of data traffic that must flow through them, i.e., $w_j^u = \frac{1}{|\mathcal{P}_u|}$. Note that when we increment the number of VOs by adding a further request, we keep fixed all network settings (i.e., s^u, d^u and $\mathcal{P}_u$ of the old VO requests) and randomly select only the parameters of the new VO.

In order to evaluate the network performance achieved by the two different optimization models that we design to compose virtual network functions, we compute the congestion experienced by the overall network and the bottleneck link, as defined in Equations (2) and (13). For each network scenario, the results we obtained represent the average of the performance metric measured over 500 network instances.

B. Performance Evaluation

Hereafter, we measure the effect of the number of virtual operators, the number of services as well as traffic demands on the performance of our proposed congestion mitigation mechanisms considering two different network topologies: the Geant and the Cogentco networks.

1) Cogentco Network Scenario: To capture the effect of the number of virtual operators and services on network congestion, we first consider the *Cogentco* network topology, and vary the number of VOs and services in the range $[10, 20]$ and $[2, 4]$, respectively. In this scenario, we fix the number of processing nodes to 2, and the VO's normalized traffic is

[2] Real network topologies can be obtained from the on-line archive maintained by the Internet Topology Zoo project, http://www.topology-zoo.org/

equal to 0.05. Parameters a_{ij} and b_{ij} are set equal to 0 and 1, respectively, for all links $(i; j)$, while $\tau = 1$.

Figures 1(a) and 1(b) show, respectively, the *total* and the *maximum* (worst) network congestion for both the MinTot- and MinMax-CMVN models, as a function of the number of virtual operators for different numbers of services in the Cogentco network scenario. Let us first discuss the effect of the number of VOs on the performance metrics.

Effect of the number of VOs: As expected, both the total congestion and the maximum congestion in the network increase with the number of VOs. Since the MinTot-CMVN model aims at minimizing the total network congestion, this performance figure is lower in MinTot-CMVN than the one obtained with MinMax-CMVN, while the maximum congestion, which is minimized by the objective function (14) of the MinMax-CMVN model shows an opposite trend. Indeed, the maximum congestion measured on a link in the considered network scenario using MinTot-CMVN is always higher ($\approx$ up to 1.5 times) than the one obtained by MinMax-CMVN, while the percentage increase in the total network congestion of MinMax-CMVN with respect to MinTot-CMVN is approximately equal to 50%.

Effect of the number of services: We now analyze the impact of the number of services on the proposed models, namely, increasing the number of services requested by each VO from 2 to 4. While Figures 1(a)-1(b) show a linear increasing relationship between the number of transmission services and the congestion (either total or worst), the flow transmitted over a link slightly change. Therefore, the link congestion is almost constant with respect to the number of transmission services, since the link capacity is evenly shared among all transmission services. This result suggests that adaptive/dynamic approaches for the assignment of the physical resources among multiple virtual services are highly recommended for resource sharing in NFV-based infrastructures.

We further observe that the computational time required to optimally solve the CMVN problem in this large-scale scenario (197 nodes and 486 directed links) is very short. Indeed, the solving time using a workstation equipped with an Intel Core (TM) Duo Processor with CPUs operating at 3 GHz and 4 Gbytes of RAM is always less than 5 s.

In order to analyze the resource utilization of the physical infrastructure, we also measured the percentage of used links when increasing the number of VOs, and fixing the number of processing nodes per VO and services to 2 and 4, respectively. The corresponding results are illustrated in Figure 11c for the MinTot-CMVN and MinMax-CMVN models. As expected, the number of used links increases almost linearly with the number of VOs, since the number of connections that are routed through the physical network increases with the number of VOs. Furthermore, the MinTot-CMVN model uses less links than MinMax-CMVN, since this latter tends to balance the VOs' traffic demands on more links in order to keep the same level of congestion on all physical links, thus removing the bottlenecks.

2) Geant Network Scenario: In this section, we evaluate the two proposed models considering the Geant network, and using the same parameters of the previous scenario: i) the number of VOs varies in the range $[10, 20]$, ii) the number of services in the range $[2, 4]$, and iii) the normalized traffic is equal to 0.05.

The Geant scenario shows similar trends, which we do not show for the sake of brevity, as those previously observed using the Cogentco network, thus demonstrating that the congestion in NFV-based infrastructures is almost/quite independent of the considered network topology. Such a result further proves the flexibility of our models and highlights some general features for congestion mitigation in virtual networks.

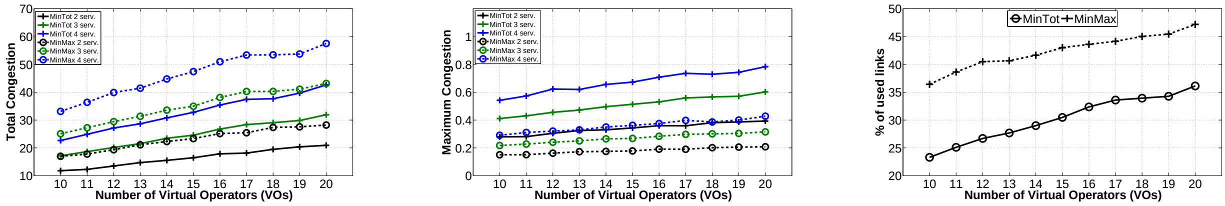
To gain further insights into the fairness of our congestion mitigation technique, we compute the empirical cumulative distribution function of the total network congestion over the simulation runs considering both uniform and skewed request distributions of 20 VOs. Corresponding results are depicted Figures 2(a) and 2(b). It can be observed that the MinTot orchestration mechanism selects a network configuration that in average results always in a lower level of total congestion, thus using more efficiently network resources than the MinMax approach. This latter uses more network resources, spreading the traffic over multiple paths to avoid high use of the bottleneck link for routing the VOs traffic through processing nodes and towards destination nodes.

Furthermore, Figure 2(a) shows that network configurations obtained using the MinMax approach exhibit larger variance on the total congestion than the corresponding solutions obtained using the MinTot optimization model. This means that the MinTot solution is less sensitive to fluctuation in the traffic distribution than the MinMax approach, thus producing more robust/stable network configurations.

It can be further observed from Figure 2(b) that the fairness of our orchestration mechanism is slightly affected by the traffic distribution. Indeed, all data connections experience approximately the same level of congestion as long as 3 transmission services are implemented, as illustrated by the empirical CDF. However, increasing the number of transmission services reduces the available bandwidth reserved for each service, which is rapidly saturated even by light connections, thus resulting in low fairness (curves corresponding to configuration with 4 services have a large dispersion of the congestion around the mean value). This result further confirm the needs of adaptive mechanisms for the assignment of physical resources among multiple virtual services.

Our numerical analysis suggests that NFV requires a *joint optimization* of all system components, including, in particular, the deployment of services within the network.

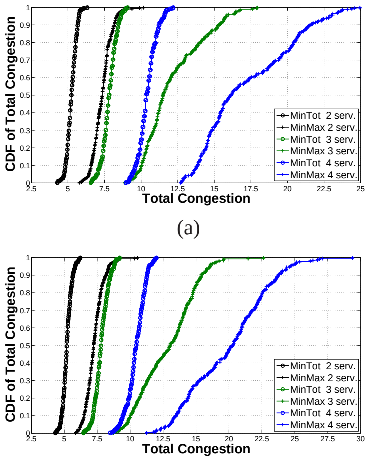
In order to gain further insights into the performance improvements achieved using our optimization models in terms of congestion reduction, we compare our approach with the heuristic approach [19], which selects the network paths using the Shorted Path Tree algorithm. Preliminary results, which we omit for the sake of brevity, show that our technique achieves a total network congestion two times lower than the value



(a) (b) (c)

Fig. 1: Network congestion (total and maximum) and percentage of used links for the MinTot-CMVN (solid line) and MinMax-CMVN models (dashed line) as a function of the number of VOs for different numbers of services in the *Cogentco* network.

measured with the SPT algorithm. Indeed, the average total congestion we measured with 4 services varies from 15 to 30 when the number of VOs increases from 10 to 20. Similarly, the SPT algorithm doubles the resource utilization, since it selects twice as many links as our solution (the number of used links varies from 70 to 95).



(a)

(b)

Fig. 2: Empirical CDF of the network congestion considering uniform and skewed distributions of VOs' requests in the *Geant* network.

VI. Conclusion

In this paper, we addressed the congestion mitigation problem in virtual networks using an optimization approach. In particular, we proposed novel orchestration mechanisms to optimally control and reduce the resource congestion of a physical infrastructure based on the NFV paradigm.

Numerical results, obtained using real ISP topologies and network scenarios, show that the proposed models can help reducing network congestion, thus representing a very practical solution to control the congestion in virtual networks.

Interestingly, our results suggest that a *joint and adaptive optimization* of all system components for NFV-based infrastructures can improve system performance. Indeed, while the separate optimization of each network function (routing, processing nodes' position) increases the flexibility of the virtual infrastructure, it ignores the increasing overhead of handling multiple services. Therefore, the adaptive assignment of physical resources among multiple virtual services represents a promising research line for NFV orchestration mechanisms.

References

[1] ETSI. Network Functions Virtualisation - Introductory White Paper. http://portal.etsi.org/NFV/NFV_White_Paper.pdf.

[2] A. Khan, A. Zugenmaier, D. Jurca, and W. Kellerer. Network virtualization: a hypervisor for the Internet? *IEEE Communications Magazine*, pages 136–143, vol. 50(1), January 2012.

[3] Q. Duan, Y. Yan, and A.V. Vasilakos. A Survey on Service-Oriented Network Virtualization Toward Convergence of Networking and Cloud Computing. *IEEE Trans. on Network and Service Management*, 9:373–392, 2012.

[4] X. Costa-Perez, J. Swetina, Tao Guo, R. Mahindra, and S. Rangarajan. Radio access network virtualization for future mobile carrier networks. *IEEE Communications Magazine*, pages 27–35, vol. 51(7), July 2013.

[5] Minlan Yu, Yung Yi, Jennifer Rexford, and Mung Chiang. Rethinking virtual network embedding: substrate support for path splitting and migration. *ACM SIGCOMM CCR*, 38(2):17–29, 2008.

[6] A. Jarray and A. Karmouch. Decomposition Approaches for Virtual Network Embedding With One-Shot Node and Link Mapping. *IEEE/ACM Trans. on Networking*, March 2014.

[7] N.M.M.K. Chowdhury, M.R. Rahman, and R. Boutaba. Virtual network embedding with coordinated node and link mapping. In *IEEE INFOCOM*, pages 783–791, 2009.

[8] X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, and J. Wang. Virtual Network Embedding Through Topology-Aware Node Ranking. *ACM SIGCOMM CCR*, 41(2):38–47, 2011.

[9] J. F. Botero, X. Hesselbach, M. Duelli, D. Schlosser, A. Fischer, and H. De Meer. Energy Efficient Virtual Network Embedding. *IEEE Comm. Letters*, 16(5):756–759, May 2012.

[10] A. Fischer, J.F. Botero, M. Till Beck, H. de Meer, and X. Hesselbach. Virtual Network Embedding: A Survey. *IEEE Communications Surveys Tutorials*, 15(4):1888–1906, 2013.

[11] M. Schöller, M. Stiemerling, A. Ripke, and R. Bless. Resilient Deployment of Virtual Network Functions. In *IEEE RNDM*, 2013.

[12] A. Jarray and A. Karmouch. VCG auction-based approach for efficient Virtual Network embedding. *IFIP/IEEE International Symposium on Integrated Network Management*, pages 609–615, 2013.

[13] Y. Xin, I. Baldine, A. Mandal, C. Heermann, J. Chase, and A. Yumerefendi. Embedding virtual topologies in networked clouds. *IEICE Future Internet Technologies*, pages 26–29, 2011.

[14] S.K. Fayazbakhsh, M.K. Reiter, and V. Sekar. Verifiable Network Function Outsourcing: Requirements, Challenges, and Roadmap. *ACM HotMiddlebox 2013*, pages 25–30, 2013.

[15] E. Altman, T. Basar, T. Jimenez, and Nahum N. Shimkin. Competitive routing in networks with polynomial costs. *IEEE Trans. on Automatic Control*, 47(1):92–96, 2002.

[16] D.P. Bertsekas and R.G. Gallagher. *Data Networks*. Prentice Hall, second edition, 1992.

[17] Bernard Fortz and Mikkel Thorup. Internet traffic engineering by optimizing ospf weights. *IEEE INFOCOM*, pages 519–528, 2000.

[18] Mosharaf Chowdhury, Muntasir Raihan Rahman, and Raouf Boutaba. Vineyard: virtual network embedding algorithms with coordinated node and link mapping. *IEEE/ACM Trans. on Networking*, 20(1):206–219, 2012.

[19] Yong Zhu and Mostafa H Ammar. Algorithms for assigning substrate network resources to virtual network components. In *IEEE INFOCOM*, pages 1–12, 2006.